

REMARKS/ARGUMENTS

Applicant responds herein to the final Office Action dated March 21, 2008.

Claims 1, 2, 4-11 and 13-15 are the claims currently pending in the present application.

Claim 1 is amended to remove a redundant instance of the word “and” connecting recitations of the claim. The amendments to the claims are believed not to raise new issues that would require further searching of the art. Accordingly, entry of the claim amendments into the record are requested at this time and the filing of a RCE is believed not to be necessary.

Rejection of Claims 1, 2, 4-11 and 13-15 under 35 U.S.C. §102

Claims 1, 2, 4-11 and 13-15 are rejected under 35 U.S.C. §102 as being anticipated by Martin (6,438,746). Reconsideration of this rejection is respectfully requested.

Claim 1 requires exception processing by detecting an occurrence of a refer request to variables arranged on a memory space managed by another processor during running of the executable form program.

As discussed in the previous Amendment, according to an aspect of applicant's invention as claimed in claim 1, after compilation of a source program into executable form programs, the executable form programs are run. Claim 1 requires that exception processing is handled by detecting an occurrence of a refer request to variables arranged on a memory space managed by another processor during running of the executable form program. After detection of the occurrence of such a refer request during running of the executable form program, the refer request is handled as described in claim 1.

Martin is directed to a pre-compiler or a compiler that reads text embedded in a comment field of source code (Martin, Abstract). Martin discloses that this comment field embedded text is read by pre-compiler 316 and is ignored by compiler 312 (Martin, column 7, lines 34-44), and that this text may include: specification data, such as data specifying the total number of objects or total corresponding volume of memory space expected for a given class; availability, or percentage of access instants on which the object must be available for access; accessibility, or the number of different access points from which the objects (or particular data within the object) may be called, or number of terminals or mainframes which will need to access the object; security, or access rights that various users are to have; concurrency, or the number of other

processes which might simultaneously be invoking a given object or process; and timeliness, or the maximum average and/or minimum time within which a process must be completed (Martin, column 7, line 50 – column 8, line 5).

Martin is silent with respect to any kind of exception handling performed during running of an executable form program, as required by claim 1. The Office Action cites Martin, column 10, lines 30-49, which discusses concurrent access to data of an object. This passage of Martin describes that, when a number of concurrent accesses to data of an object are possible, an inconsistent situation can arise and when pre-compiler 316 reads that for a class a number of concurrent processes may access an object of the class, then in step 5006 of Fig. 13 a concurrency control method is selected for that class, which may be a simple locking process whereby, after an object has been accessed, no further access is possible until the first process has terminated (Martin, column 10, lines 30-46). Thus, Martin provides that pre-compiler 316 uses conventional means for locking access to an object if multiple accesses to the object are possible.

Martin does not disclose or suggest exception processing by detecting an occurrence of a refer request to variables arranged on a memory space managed by another processor during running of the executable form program, as required by claim 1.

In fact, Martin is completely silent with respect to handling references to variables arranged on a memory space managed by another processor, as required by claim 1. The Office Action cites Martin, column 9, lines 26-33 and alleges that this passage discloses “the same memory space.” As discussed, Martin discloses that pre-compiler 316 reads text contained in the comment field, this text including size statements associated with classes so that memory space of a suitable size (in terms of megabytes) may be set aside. However, this has nothing to do with exception handling for references to variables arranged on a memory space managed by another processor, as recited in claim 1.

More generally, as shown in Fig. 6 of Martin, pre-compiler phase 316, during which the text embedded in the comment fields are read, and compiler phase 312 occur before the linker phase and the later phases, and thus, Martin is silent with respect to the types of exception processing for run time operations recited in claim 1. Accordingly, Martin does not disclose or suggest the recitations of claim 1.

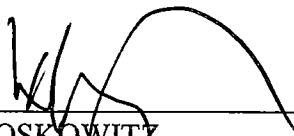
Claims 2, 4-11 and 13-15 depend from claim 1 and are therefore patentably distinguishable over the cited art for at least the same reasons.

In view of the foregoing discussion, withdrawal of the rejection and allowance of the claims of the application are respectfully requested.

Accordingly, the Examiner is respectfully requested to reconsider the application, allow the claims as amended and pass this case to issue.

THIS CORRESPONDENCE IS BEING
SUBMITTED ELECTRONICALLY
THROUGH THE UNITED STATES
PATENT AND TRADEMARK OFFICE
EFS FILING SYSTEM
ON JUNE 23, 2008

Respectfully submitted,



MAX MOSKOWITZ
Registration No.: 30,576
OSTROLENK, FABER, GERB & SOFFEN, LLP
1180 Avenue of the Americas
New York, New York 10036-8403
Telephone: (212) 382-0700